UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/631,308	07/31/2003	Gerard Chauvel	TI-35433 (1962-05412)	1887

23494 7590 07/25/2006

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
PETRANEK, JACOB ANDREW	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 07/25/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/631,308	**Applicant(s)** CHAUVEL ET AL.	
	Examiner Jacob Petranek	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 June 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16,18-27 and 30-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16,18-27 and 30-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-16, 18-27, and 30-41 are pending.

2. The office acknowledges the following papers:

Specification, claims, and arguments filed on 6/23/2006.

Withdrawn objections

3. The specification objections have been withdrawn.

New Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in-
> (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effect under this subsection of a national application published under section 122(b) only if the international application designating the United States was published under Article 21(2)(a) of such treaty in the English language; or
> (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that a patent shall not be deemed filed in the United States for the purposes of this subsection based on the filing of an international application filed under the treaty defined in section 351(a).

5. Claims 1-11 are rejected under 35 U.S.C. §102(e) as being anticipated by Patel (U.S. 6,826,749).

6. As per claim 1:

Patel disclosed a processor, comprising:

A multi-entry stack usable in at least a stack-based instruction set (Patel: Figure 3 element 50, column 5 lines 14-26)(The stack is used for the stack-based instructions.);

Logic coupled to said stack, the logic manages the stack (Patel: Figure 4 element 74, column 6 lines 43-56)(The state machine updates the stack and the pointers to the stack.); and

A plurality of registers coupled to the logic and addressable through a second instruction set that provides register-based and memory-based operations (Patel: Figure 3 elements 46 and 48, column 6 lines 57-61); and

An instruction fetch logic that receives at least stack-based instructions from the stack-based instruction set (Patel: Figure 4 element 70, column 6 lines 18-33)(Element 70 is an instruction buffer that receives stack-based instructions from the instruction cache. It's inherent that there's fetch logic to fetch these stack-based instructions from the instruction cache.).

7. As per claim 2:

Patel disclosed the processor of claim 1 wherein the stack has a top and the stack is accessible within the second instruction set through at least one of the registers in which a value is stored that is present at the top of the stack (Patel: Figure 3 element 44, column 5 lines 14-26; Figure 5, column 7 lines 31-54)(The java registers store a pointer to the top of the stack and has a register storing the value.).

8. As per claim 3:

Patel disclosed the processor of claim 1 wherein the stack has a top that is addressable by a memory mapped address, and the memory mapped address is stored in a register which is accessed by the second instruction set (Patel: Figure 3 element

44, column 5 lines 14-26; Figure 5, column 7 lines 31-54)(The register stores a pointer that points to the top of the stack. Thus having the same functionality.).

9. As per claim 4:

Patel disclosed the processor of claim 1 wherein the stack-based instruction set accesses operands from the stack and places results from operations on the stack and, as a result of accessing operands the stack and placing results on the stack, at least some of the registers are updated (Patel: Figure 3 element 44 and 50, column 6 lines 43-56; Figure 5, column 7 lines 30-54)(The pointer register and register holding the top value of the stack are updated each time the stack changed. Thus having the same functionality.).

10. As per claim 5:

Patel disclosed the processor of claim 1 further comprising a first program counter usable in the execution of the stack-based instruction set and a second program counter usable in the execution of a micro-sequence that comprises instructions from both the stack-based and second instruction sets (Patel: Figure 3 elements 54 and 44, column 5 lines 33-40).

11. As per claim 6:

Patel disclosed the processor of claim 1 further comprising a pair of parallel address generation units coupled to said logic which are used to compute memory source and destination addresses and wherein a register includes the top of the multi-entry stack, thereby permitting a block of data to be moved between a memory area and the stack by execution of a single instruction with a repeat loop (Patel: Figure 3 elements 24 and 58, column 5 lines 60-67 continued to column 6 lines 1-3).

12. As per claim 7:

Patel disclosed the processor of claim 1 wherein the second instruction set comprises an instruction that retrieves operands from memory, performs a computation on the operands, and places the result on the stack (Patel: Figure 3 elements 58 and 50, column 5 lines 60-67 continued to column 6 lines 1-3)(A load/store instruction will retrieve and store operands to/from memory. Pop and push instructions will load and store operands to/from a stack.).

13. As per claim 8:

Patel disclosed a method of processing instructions in a processor, comprising:

Fetch logic receiving instructions from a first instruction set which comprises stack-based instructions (Patel: Figure 4 element 70, column 6 lines 18-33)(Element 70 is an instruction buffer that receives stack-based instructions from the instruction cache. It's inherent that there's fetch logic to fetch these stack-based instructions from the instruction cache.).

Fetch logic receiving instructions from a second instruction set which comprises memory-based and register-based instructions (Patel: Figure 1 element 28, column 3 lines 58-67 continued to column 4 lines 1-9)(The CPU fetches and executes Java and non-Java code. Thus having the same functionality.); and

Executing said received instructions from the first and second instruction sets (Patel: Figure 1 element 26, column 3 lines 58-67 continued to column 4 lines 1-9)(The CPU fetches and executes Java and non-Java code. Thus having the same functionality.);

14. As per claim 9:

The method of claim 8 further comprising forming a sequence of instructions from both of said first and second instruction sets (Patel: Figure 3 element 28, column 6 lines 18-33)(Instructions that are to be executed can be from both ISA's.).

15. As per claim 10:

Claim 10 essentially recites the same limitations of claim 7. Therefore, claim 10 is rejected for the same reasons as claim 7.

16. As per claim 11:

Claim 11 essentially recites the same limitations of claim 4. Therefore, claim 11 is rejected for the same reasons as claim 4.

New Claim Rejections - 35 USC § 103

17. The following is a quotation of 35 U.S.C. §103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

18. Claims 12-16, 18-24, and 26-27 are rejected under 35 U.S.C. §103(a) as being unpatentable over Patel et al. (U.S. 6,826,749), in view of Hennessy et al. ("Computer Architecture: A Quantitative Approach").

19. As per claim 12:

Claim 12 essentially recites the same limitations of claim 1. Claim 12 additionally recites the following limitations:

Patel disclosed memory coupled to said logic (Patel: Figure 3 element 58); and

Wherein a first register includes an address through which the top of the stack is accessed and a second register in which a value at the top of the stack is stored (Patel: Figure 3 element 44, column 5 lines 14-26; Figure 5, column 7 lines 30-54)(Register 1 stores the value at the top of the stack.).

Patel failed to teach wherein at least one of the registers are used to calculate addresses in parallel, said addresses being calculated in accordance with any of a plurality of addressing modes specified by the second instruction set.

However, Hennessy disclosed wherein at least one of the registers are used to calculate addresses in parallel, said addresses being calculated in accordance with any of a plurality of addressing modes specified by the second instruction set (Hennessy: Figures 2.6 and 2.7, section 2.3)(Register addressing mode and register indirect modes for load and stores use a register for calculating the effective address.).

Patel disclosed data memory accesses to the data cache through load and store instruction, but left out the details of the addressing modes used to calculate the memory address needed to access the data or storing the data within the data cache (Patel: Figure 3 element 58, column 5 lines 60-67 continued to column 6 lines 1-3). One of ordinary skill in the art would have been motivated to look for different types of addressing modes that could be used to load and store data from memory. Hennessy disclosed many different types of addressing modes for data memory instructions.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to add the addressing modes of Hennessy to Patel's load and store instructions.

20. As per claim 13:

Patel and Hennessy disclosed the processor of claim 12 wherein the stack-based instruction set accesses operands from the stack and places results from operations on the stack and, as a result of accessing operands from the stack and placing results on the stack thereby causing the address in the first register to be changed (Patel: Figure 3 element 44 and 50, column 6 lines 43-56; Figure 5, column 7 lines 30-54)(The pointer register and register holding the top value of the stack are updated each time the stack changed. Thus having the same functionality.).

21. As per claim 14:

Patel and Hennessy disclosed the processor of claim 13 wherein the address in the first register is incremented or decremented depending on whether the register is used as a source or a destination, respectively, for an operation (Patel: Figure 3 elements 44 and 50, column 6 lines 43-56; Figure 5, column 7 lines 30-54)(The pointer is incremented or decremented depending on if the top value is popped off the stack or if a value is pushed on the stack. Thus having the same functionality.).

22. As per claim 15:

Patel and Hennessy disclosed the processor of claim 12 wherein the stack-based instruction set comprises Java Bytecodes (Patel: Column 4 lines 33-46).

23. As per claim 16:

Claim 16 essentially recites the same limitations of claim 5. Therefore, claim 16 is rejected for the same reasons as claim 5.

24. As per claim 18:

Patel and Hennessy disclosed the processor of claim 12 wherein at least one of the registers includes an offset usable in the calculation of addresses (Hennessy: Figures 2.6 and 2.7, section 2.3)(Register indirect or indexed addressing modes use registers as pointers to the effective address. Displacement addressing mode uses an immediate value to add to a register value for the effective address.).

25. As per claim 19:

Patel and Hennessy disclosed the processor of claim 12 wherein the second instruction set comprises an instruction that moves data from a register or memory to a register, and consequently to the stack (Patel: Figure 3 element 58, column 5 lines 60-67 continued to column 6 lines 1-3)(Load/store instructions move data to/from the data cache. Thus having the same functionality.).

26. As per claim 20:

Patel and Hennessy disclosed the processor of claim 19.

Wherein the instruction that moves data includes a plurality of bits of that encode one of a plurality of addressing modes (Hennessy: Figures 2.6 and 2.7, section 2.3).

27. As per claim 21:

Patel and Hennessy disclosed the processor of claim 20 wherein the addressing modes include a mode in which the instruction that moves data includes an immediate value and a reference to a register containing a base address, wherein the immediate value and the base address are added together to generate a source memory address for the move instruction (Hennessy: Figures 2.6 and 2.7, section 2.3)(Displacement addressing mode uses an immediate value that's added to the value contained within a register to obtain the effective address.).

28. As per claim 22:

Claim 22 essentially recites the same limitations of claim 21. Therefore, claim 22 is rejected for the same reasons as claim 21.

29. As per claim 23:

Patel and Hennessy disclosed the processor of claim 20 wherein the addressing modes include a mode in which the instruction that moves data includes references to two registers in which memory addresses are stored, one register being a predetermined index register, the memory addresses from the two registers are added together to calculate a source memory address used to complete the move instruction, and the address in the predetermined index register is incremented (Hennessy: Figure 14, column 11 lines 40-67 continued to column 12 lines 1-26)(Autoincrement adds two register values and increments the index register.).

30. As per claim 24:

Claim 24 essentially recites the same limitations of claim 23. Therefore, claim 24 is rejected for the same reasons as claim 23.

31. As per claim 26:

Claim 26 essentially recites the same limitations of claim 5. Therefore, claim 26 is rejected for the same reasons as claim 5.

32. As per claim 27:

Claim 27 essentially recites the same limitations of claim 6. Therefore, claim 27 is rejected for the same reasons as claim 6.

33. Claim 25 is rejected under 35 U.S.C. §103(a) as being unpatentable over Patel et al. (U.S. 6,826,749), in view of Hennessy et al. ("Computer Architecture: A Quantitative Approach"), further in view of Hendler et al. (U.S. 6,473,777) and Brassac et al. (U.S. 6,928,539).

34. As per claim 25:

Patel and Hennessy disclosed the processor of claim 12.

Patel and Hennessy failed to teach wherein the processor is configured to be coupled to a separate processor on which an operating system is executed.

However, Hendler disclosed wherein the processor is configured to be coupled to a separate processor (Hendler: Figure 2 element 102, column 2 lines 32-46)(Element 102 executes various overhead activities associated with java instructions. Thus having the same functionality.).

The advantage of having another processor to run processes is that it can run overhead tasks, such as compilation and garbage collection, which needs to be done to execute Java instructions (Hendler: Column 1 lines 44-64). The processor running Java instruction will make gains in performance by having another processor handler these overhead tasks (Hendler: Column 1 lines 44-64). The performance increase of a java stack-based processor would have motivated one of ordinary skill in the art to use another processor for overhead tasks. Thus, it would have been obvious to one of ordinary skill in the art to add another processor to Patel's system that deals with executing overhead tasks for the advantage of increased performance in the java processor.

Patel, Hennessy, and Hendler failed to teach a separate processor on which an operating system is executed.

However, Brassac disclosed a separate processor on which an operating system is executed (Brassac: Figure 2 element 8, column 1 lines 58-67 continued to column 2 lines 1-7)(The separate processor runs operating system tasks for other processors. Thus having the same functionality.).

The advantage of having another processor to run processes is that it can run overhead tasks, such as compilation and garbage collection, which needs to be done to execute Java instructions (Hendler: Column 1 lines 44-64). Executing instructions for the operating system is another such task that can be considered an overhead task. Running the operating system on the separate processor would have increased the performance of the java processor. One of ordinary skill in the art would have been motivated by this increased performance of the java processor to have the separate processor run OS instructions. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a separate processor running OS instructions for the added performance to the java processor.

35. Claims 30-41 are rejected under 35 U.S.C. §103(a) as being unpatentable over Patel et al. (U.S. 6,826,749), in view of Hendler et al. (U.S. 6,473,777).

36. As per claim 30:

Patel disclosed a system, comprising:

A co-processor comprising a stack, fetch logic, and registers, said fetch logic receiving stack-based instructions from a first instruction set (Patel: Figure 4 element 70, column 6 lines 18-33)(Element 70 is an instruction buffer that receives stack-based instructions from the instruction cache. It's inherent that there's fetch logic to fetch these stack-based instructions from the instruction cache.), said co-processor is configured to execute stack-based instructions from a first instruction set and instructions from a second instruction set that provides memory-based and register-based operations (Patel: Figure 1 elements 26 and 28, column 3 lines 58-67 continued to column 4 lines 1-9; Figure 3 elements 44, 46 and 58, column 5 lines 14-26)(The co-processor executes stack-based and register-based instructions using a stack and register file to execute the instructions. The co-processor also executes memory instructions for getting data from the data cache.).

Patel failed to teach a main processor unit coupled to the co-processor.

However, Hendler disclosed a main processor unit (Hendler: Figure 2 element 102, column 2 lines 32-46)(Element 102 executes various overhead activities associated with java instructions. Thus having the same functionality.).

The advantage of having another processor to run processes is that it can run overhead tasks, such as compilation and garbage collection, which needs to be done to execute Java instructions (Hendler: Column 1 lines 44-64). The processor running Java instruction will make gains in performance by having another processor handler these overhead tasks (Hendler: Column 1 lines 44-64). The performance increase of a java stack-based processor would have motivated one of ordinary skill in the art to use another processor for overhead tasks. Thus, it would have been obvious to one of ordinary skill in the art to add another processor to Patel's system that deals with executing overhead tasks for the advantage of increased performance in the java processor.

37. As per claim 31:

Claim 31 essentially recites the same limitations of claim 15. Therefore, claim 31 is rejected for the same reasons as claim 15.

38. As per claim 32:

Patel disclosed The system of claim 31 further including a compiler coupled to said co-processor, said compiler receives Java bytecodes and replaces at least one group of bytecodes by a sequence of instructions from the second instruction set and provides said sequence to the co-processor for execution (Patel: Column 1 lines 34-43 and column 4 lines 33-46).

39. As per claim 33:

The system of claim 32 wherein the sequence also includes stack-based instructions from the first instruction set (Patel: Column 1 lines 34-43 and column 4 lines 33-46).

40. As per claim 34:

Patel disclosed the system of claim 30 wherein the system comprises a cellular telephone (Official notice is given that the processor of Patel and Hendler could be put in a cell phone.).

41. As per claim 35:

Claim 35 essentially recites the same limitations of claim 2. Therefore, claim 35 is rejected for the same reasons as claim 2.

42. As per claim 36:

Claim 36 essentially recites the same limitations of claim 3. Therefore, claim 36 is rejected for the same reasons as claim 3.

43. As per claim 37:

Claim 37 essentially recites the same limitations of claim 4. Therefore, claim 37 is rejected for the same reasons as claim 4.

44. As per claim 38:

Claim 38 essentially recites the same limitations of claim 5. Therefore, claim 38 is rejected for the same reasons as claim 5.

45. As per claim 39:

Patel disclosed the system of claim 38 wherein the first and second program counters are stored in said registers (Patel: Figure 3 elements 44 and 54, column 5 lines 33-40).

46. As per claim 40:

Claim 40 essentially recites the same limitations of claim 5. Therefore, claim 40 is rejected for the same reasons as claim 5.

47. As per claim 41:

Claim 41 essentially recites the same limitations of claim 6. Therefore, claim 41 is rejected for the same reasons as claim 6.

Response to arguments

48. The arguments presented by Applicant in the response, received on 6/23/2006 are partially considered persuasive.

49. Applicant argues that "Patel doesn't have instruction fetch logic that receives stack-based instructions."

This argument is not found to be persuasive for the following reason. The examiner agrees that the IFU of the CPU doesn't receive stack-based instructions from the instruction cache, because they are translated by the time they reach the IFU. However, the translator itself contains instruction fetch logic that receives stack-based instructions, java bytecodes, and translates them into native instructions. Element 70 is an instruction buffer that receives stack-based instructions from the instruction cache. It's inherent that there's fetch logic to fetch these stack-based instructions from the instruction cache.).

50. Applicant argues that "The Balmer (U.S. 6,839,831) reference cannot be used against claim 12 in a 103 rejection because it's commonly owned."

This argument is found to be persuasive for the following reason. A new ground of rejection has been given due to the amendment.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob Petranek whose telephone number is 571-272-5988. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jacob Petranek
Examiner
Art Unit 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100